# 8085 Addressing Modes and Memory Mapping

# Addressing Modes of 8085

The various formats for specifying operands are called addressing modes. The different ways that a microprocessor can access data. The way the operands are chosen during execution of an instruction is determined by addressing modes.

Different addressing modes of 8085 microprocessor –

i. Immediate addressing mode
ii. Register addressing mode
iii. Direct addressing mode
iv. Indirect addressing mode
v. Implicit/Implied addressing mode

## i. Immediate addressing mode:

8 or 16 bit data can be specified as a part of instruction. Data is present in the instruction itself. Whenever the symbol I is present in the instruction then it is an immediate addressing mode.

Example: MVI B, 33H; Move immediate 33H data into B register

MVI M, 7CH; 7CH is copied into memory "M"

LXI D, 245EH; 24H is copied to D and 5E is copied to E register

ADI 87H; 87+A → A

## ii. Register addressing mode:

Data transfer between register specifies the source, destination and both operands in 8085 register.

Example: MOV A,B; the content of B will be move to A register

ADD E; E+A → A

SP H, L; contents of H and L will be store in the top of the stack.

### iii. Direct addressing mode:

It specifies 16-bit address of the operand within instruction itself.

Example: LDA 6000H; load accumulator with data present at 6000H

### iv. Indirect addressing mode:

Address of the data is present as content of another register pair. The data is transferred from the address pointed by the data in a register to other register.

Example: MVI M, 55H; Immediate Indirect

LDAX D; Load accumulator with data whose address is present at D E register pair

### v. Implicit/Implied addressing mode:

This mode doesn't require any operand. Opcode specifies the address of operand. But some kind of operation is performed by the instruction. And two operations may be performed in single register.

Example: CMA; Complement the content of Accumulator (70 8F) →

STC; Set Carry Flag RAL; Rotate Accumulator Left

## Determine the starting address of 4KB memory with ending address BA3F H.

### Solution:

Starting address = Ending address – OFFSET

Ending address = Starting address + OFFSET

OFFSET = Ending address when starting address is 0000 H

4KB = 4 * 1KB = 4 * 1024 * 8 = 2^2 * 2^10 * 8 = 2^12 * 8

= 2^x * y (Size of ROM), where x is the number of address lines and y is the number of data lines

0000 H = 0000 0000 0000 0000 (Starting address)

0FFF H = 0000 1111 1111 1111 (Ending address)

2KB = 2 * 1KB = 2 * 1024 * 8 = 2^1 * 2^10 * 8 = 2^11 * 8

0000 H = 0000 0000 0000 0000 (Starting address)

07FF H = 0000 0111 1111 1111 (Ending address)

Now, Starting address = Ending address – OFFSET

= BA3F – 0FFF

= AA40 H (Answer)

**Determine the size of memory whose starting and ending address are 4A00 H and 69FF H respectively.**

**Solution:**

Size of memory = Ending address – Starting address

= 69FF – 4A00 = 1FFF H

Now, 1FFF H = 0001 1111 1111 1111

(Thirteen 1s are there. These 1s represent the number of address lines. Data lines are 8)

So, Memory size = 2^x * y (Where, x = Number of address lines and y = Number of data lines)

= 2^13 * 8 = 2^3 * 2^10 * 8 (Since, 2^10 * 8 = 1 KB)

= 2^3 * 1KB = 8 KB **(Answer)**

## Memory Mapping :

It is the process of assigning address range to each memory IC in a microcomputer.

**Problem:** An 8-bit microprocessor has 16-bit address bus (A0 – A15) with a 1KB memory chip as shown in the figure. What is the address range for the chip?

| A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **(F000 H)** |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | **(F3FF H)** |